**MEAN Stack implemented!!!**

**This application covers following USE CASES:**

1. Signup/Registration feature.
2. Login feature.
3. Angular form validations are available.
4. After registration and login, user is allowed to see a dashboard.
5. Dashboard is visualizing some dummy data by displaying twoD3js(NVD3) charts.
6. User can logout and is redirected to login page again.
7. Login security implemented properly as without login user is not able to see the dashboard directly by typing URL.
8. Login/registration data and data for D3-charts are stored onMongoDB and served by NodeJS server by create and read operation.
9. Application is completely responsive.
10. I have not paid attention on data security over the network.

**Instructions to run the app:**

1. Install NodeJS.
2. Copy the attached application on some place, let’s say as D:/MEAN-Stack-App/
3. Download MongoDB .msi file and install MongoDB.
4. Open command prompt and run following command to start mongoDB server and point to database folder (I have kept related database on MEAN-Stack-App/app/data):

cd path/To/MongoDB/Bin

mongod –dbpath D:/MEAN-Stack-App/app/data.

1. Open another command prompt and start NodeJS server by running following command :-

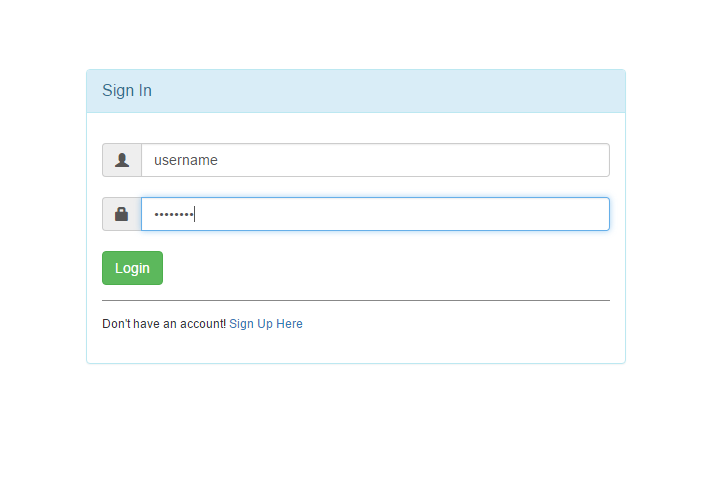
cd D:/MEAN-Stack-App/

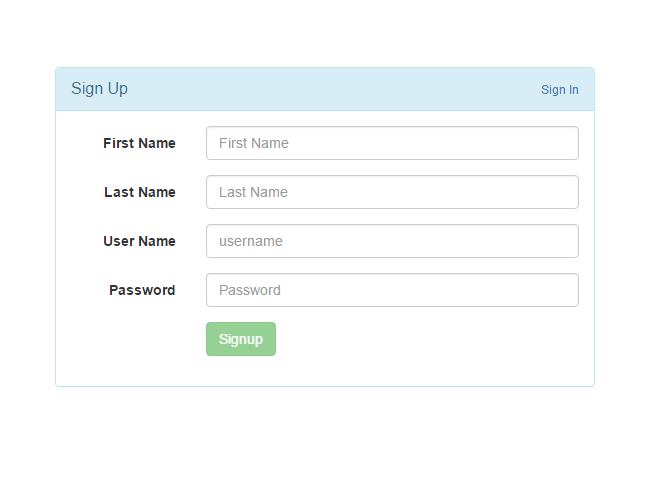
nodeserver-express-mongo.js

1. Open browser and hit the URL <http://localhost:8000/>
2. Now we have a running MEAN-Stack application. Just use it ☺

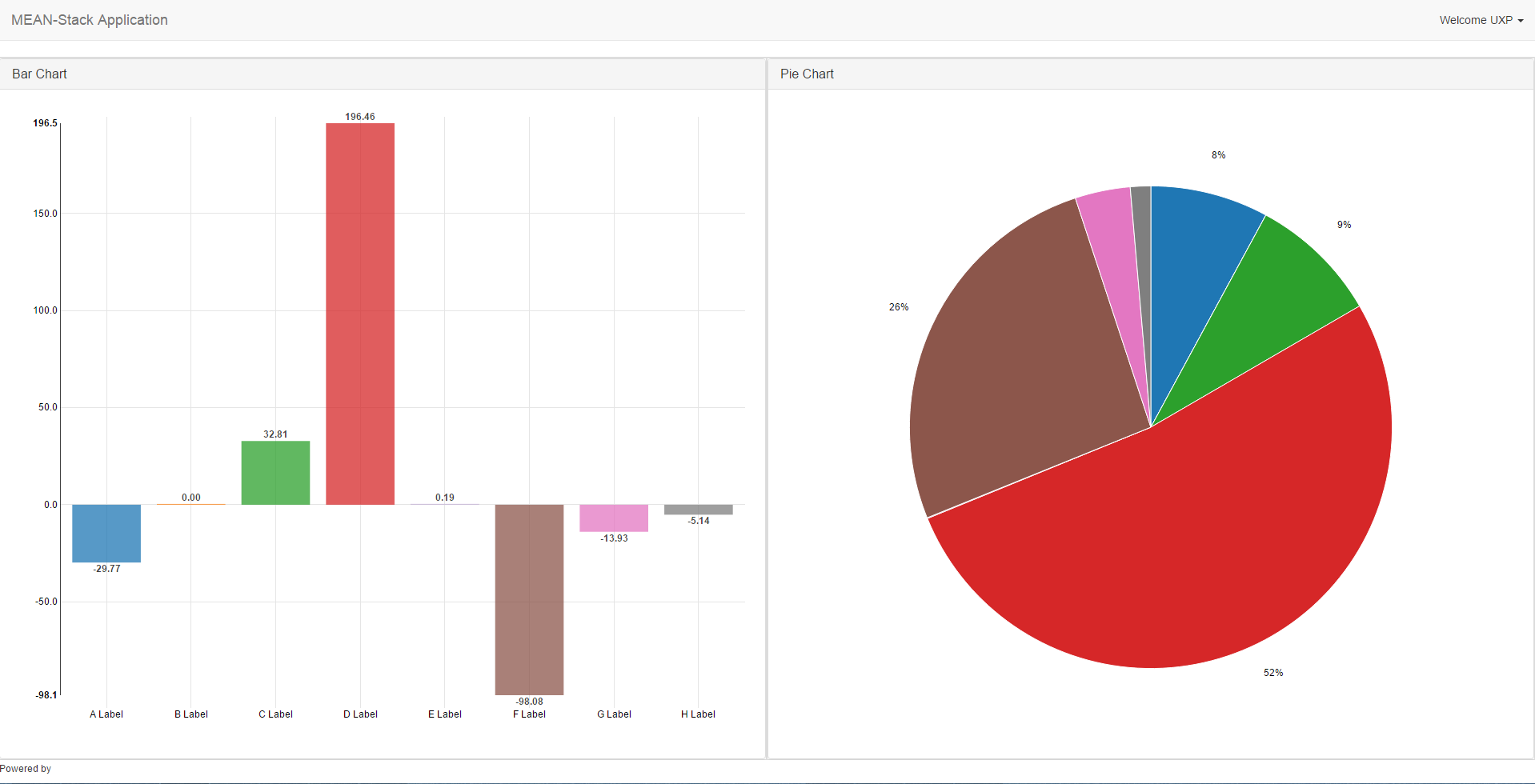
Snapshots of application/POC:

Login form:

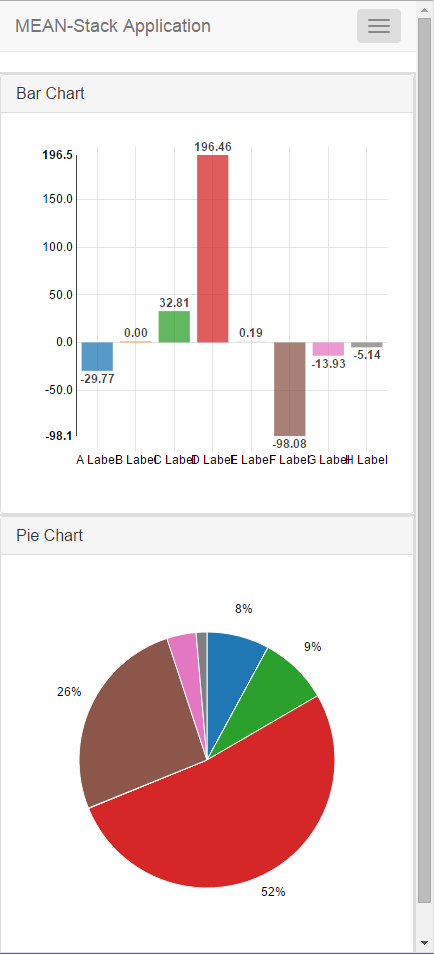


Registration form: 

Dashboard:



Responsiveness:



Imp Commands:

> db.project.update({"projectTitle":"test"},{$push:{"statusArray":"dddd"}})

> db.project.update({"projectTitle":"test"},{$push:{"statusArray":"test"}})

Added to status array

{

"\_id" : ObjectId("

"projectTitle" : "

"projectDesc" : "s

"projectTeam" : [

"statusArray" : [

"test",

"dddd"

]}

db.project.update({"projectTitle":"test"},{$pull:{"statusArray":{$in : ["ddd"]}}})

db.project.update({"projectTitle":"test"},{$pull:{"statusArray":{$in : ["test"]}}})

resulted in

{

"\_id" : ObjectId("

"projectTitle" : "

"projectDesc" : "s

"projectTeam" : [

"statusArray" : [

"test",

"dddd"

]}

For embedded

db.project.update({"projectTitle":"test",},{$push:{"statusArray":{"datate" : "1/3/32"}}})

adds to status array and object

db.project.update({"projectTitle":"test"},{$pull:{"statusArray":{"datate" : "1/3/32"}}})

removes from status array and object